BRNO UNIVERSITY OF TECHNOLOGY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF MATHEMATICS

IMAGE ANALYSIS IN HEAT TRANSFER

DIPLOMA THESIS

AUTHOR MILAN HNIZDIL

SUPERVISOR Prof. JAROSLAV HORSKY, Ph.D.

July 1, 2008

Abstract
The object of this work is to describe and compare water tracks of high pressure nozzles and explain image processing methods to the reader. Mathematical methods used in the program for getting properties of different jets are also present.

key words
Fourier transform, Fourier series, Image analysis, Two dimensional signals, FastFourier transform.

Milan Hnizdil: *Image analysis in heat transfer*, Brno, University of technology, Faculty of mechanical engineering, 2008 (52 pages). Supervisor Prof. Jaroslav Horsky, Ph.D.

I proclaim that i have elaborated my thesis *Image analysis in heat transfer* self-containedly under Prof. Jaroslav Horsky, Ph.D., using materials written in the bibliography.

Milan Hnizdil

I would like to thank Prof. Druckmüller and all people in Fluid flow and heat transfer laboratory for their help with my work and their support.

Milan Hnizdil

Contents

Introduction

In the second half of the 20th century, digital photography was developed. Scientists started to use mathematical methods for the processing of images. These numerical methods in image analysis form the foundations of many branches, for example: medicine, telecommunications, defense, astronomy, material science, security, robotics and so on.

The most important method is the discrete fast Fourier transform of two-dimensional data, or signals (2-D DFT). With the evolution of telecommunication, the one dimensional Fourier transform was used for repressing noise. The father of cybernetics, Norbert Wiener[1], demonstrated the dependence between linear filtration and the Fourier transform. In 1965, mathematicians Cooley and Tukey[2] discovered the fast Fourier transform algorithm for digital signals. The Fourier transform and Fourier series are a very difficult and extensive topic. For this reason, the third, fourth, fifth chapter are composed more practically and follows the publication of Václav Bezvoda et al. called Dvojrozměrná diskrétní Fourierova transformace a její použití.

The first chapter consists of an introduction to image analysis. The second chapter explains the problematic of the discrete and continuous signal. The third, fourth and fifth chapters deal with the Fourier transform, Fourier series, their properties and relations between them. At the end of the work, the used program is described including its functions.

[1] American mathematician born in 1894

[2] American mathematicians, Dr. John Cooley born in 1926 and John Wilder Tukey born in 1915

Chapter 1

Image

The image is represented by a matrix A containing real numbers. For the black and white[1] picture, one number represents a point in the image, respectively pixel. In the coloured image, one pixel is represented by four numbers (RGB or CMYK)[2]. The colour of each pixel is created by the composition of these numbers.

1.1 Visualization

Each digital camera contains a kernel called CCD or CMOS chip. The intensity of the light going through the optical system to the chip is transformed to electrical charge. This charge is measured for each point of the image. The obtained signal is transferred to binary code and converted to image data, for example RAW, JPEG and so on.

1.1.1 CCD chip

The CCD chip is composed from some small semiconducting elements which are free before the light lands. During the exposition time, photons are going directly through the optical system and filters to these small elements of the chip. This chip is divided into cells which all contain 4 parts with different filters. Each filter transmists just an individual colour (red, green, blue and green). The intensity of each colour is transfered by semiconducting elements to the electrical charge. The brighter spaces produce more charges than the darker places. So each element has a new value of the charge and it is possible to compute and digitize the signal.

1.1.2 CMOS chip

The CMOS chip is composed from some small semiconducting elements as the CCD chip, but they are controlled by the electrical field. So the field generates the electrical charge, which is proportional to the intensity of the light incident on this element. The charge is digitized, however the signal contains noise. So each element contains the analytic circuit which measures and eliminates the noise. The CMOS technology is used in micropro-

[1]Usually called black and white, actually they are grey shades
[2]Red, green, blue or cyan, magenta, yellow and black

cessors, microcontrollers and other digital logic circuits. As a consequence, it is not so expensive to produce CMOS chips and they do not spend as much energy as CCD chips.

1.1.3 RGB

The human eye indicates different wavelengths of light. It responds mostly to yellow, green and violet colours[3] which gives an RGB triangle[4], see figure 1.1. This model describes how much of each colour is included on the black background and is used in most screens, cameras and video cameras in the World. When the screen is switched off, it just mean that basic colours are equal to 0. If there is just a white colour, it means that the colours are equal to the maximal value of the relevant data type.

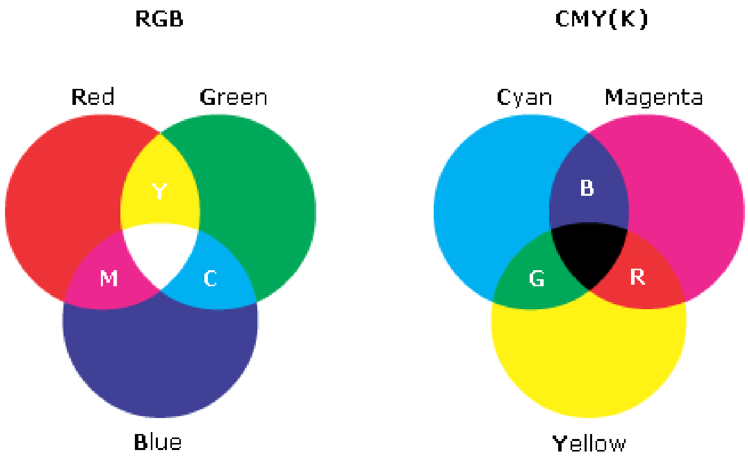

Figure 1.1: RGB model on the left and CMYK model on the right

1.1.4 CMYK

The CMYK model is used mostly for printing. It is the inverse procedure to the RGB model. It means that by applicating maximum values of the first three basic colours on a white background (paper), we obtain the black colour, see figure 1.1. It is not needed to use the black colour for printing, but there are two reasons to use it directly then mix it by three others basic colours. It is cheaper and by mixing all three basic colours we do not obtain black, but very dark grey colour.

1.2 Data types

Pictures are saved in different data types. If the matrix is composed from the numbers between 0 and 255, it is called an 8 bite image. This type is used just for viewing with the human eye. Ordinary eyes can differentiate around 150 degrees of brightness. For this reason, the 8 bite type is sufficient for observation by human eye, however it is not sufficient for mathematical processing.

[3]The violet colour is replaced by green
[4]The fourth position is represented by the green colour or it is just free

The next type is called Word type. The matrix includes numbers from the interval $\langle\, 0\,;\, 65535\,\rangle$. This format is nonsensical for viewing by the human eye. But it is extremely important for mathematical processing. This format is also called the 16 bite type.

1.3 Image format

All companies use secret methods to store images to memory cards[5]. For compatibility with other programs, it is saved to known predefined formats. As follows, the created format tells the other program how to work with the picture.

1.3.1 RAW format

Data saved to the RAW format contains only raw data. It is not possible to view an image of this format without further processing. Each company has their own RAW type. For example Nikon has nef, Canon use crv or cr2 and Olympus use orf. For opening raw file, the user needs a program which can read data from this file. It has to perform many procedures to open a raw file, for example colour correction, reduction of noise, creating colour from each four coloured positions on the chip etc.

1.3.2 JPEG format

The most often used image format is the 8 bite jpg. It uses the Fourier transform method, see chapter 3.1, which separates spatial frequencies. Our eye is sensitive to high frequencies and practically does not see low frequencies. The Fourier transform detects low frequencies and removes them. The new size of the image file is ten times smaller than the original one. This method is one example of loss compress methods which means that the user is not able to return the original picture.

1.3.3 FTS and FIT format

These two formats are created by the Iris astronomy program. This program is freeware[6] and can transform data from the raw file to the matrix containing information about the image (e.g image width, height and so on) and colours for all pixels. It saves the data to the fit or fts format. These data are not compressed and are useful for the following mathematical processing.

1.4 Noise

Noise is a random image component. There exist many kinds of noise, which arise for example by the height temperature of the cameras components, by errors during writing and reading digital data, by defective sensors, by dust or by collisions energy elements[7] with the ccd chip. There exist many methods of filtering the noise, methods which can detect it and correct data. However, this is not the subject of this work.

[5] Consider digital cameras
[6] Software for free
[7] Photons or gamma radiation

Chapter 2

2-D Signals

Image processing requires certain data[1]. This data is referred to as signals. Two basic types of signals are continuous and discrete signals. The continuous signal is a function of two dimensional real variables, while the discrete signal is represented by function of two dimensional integer variables.

2.1 Continuous signal

A continuous signal is a real or complex function $s(t_1, t_2)$ of two real variables $t_1,\ t_2$, whose domain is the real plane R^2.

The closure of the domain subset where the continuous signal gets non-zero values is called the signal support. As follows, the continuous signal $s(t_1, t_2)$ is spatially limited in the interval $\langle Q_1, Q_2 \rangle$, see figure 2.1.

$$s(t_1, t_2) \neq 0 \qquad 0 \leq t_1 \leq Q_1, \quad 0 \leq t_2 \leq Q_2, \tag{2.1}$$

$$s(t_1, t_2) = 0 \qquad elsewhere \tag{2.2}$$

The support of this continuous signal is the rectangle

$$P_{Q_1,Q_2} = \Big\{ (t_1, t_2) \in R^2 : \ 0 \leq t_1 \leq Q_1,\ 0 \leq t_2 \leq Q_2 \Big\}. \tag{2.3}$$

A periodic continuous signal is a continuous signal $\tilde{s}(t_1, t_2)$, for which there exists a pair of real numbers (T_1, T_2) such that

$$\tilde{s}(t_1 + T_1, t_2) = \tilde{s}(t_1, t_2), \tag{2.4}$$

$$\tilde{s}(t_1, t_2 + T_2) = \tilde{s}(t_1, t_2) \tag{2.5}$$

for all $(t_1, t_2) \in R^2$. The couple (T_1, T_2) is called the periodicity of the signal $\tilde{s}(t_1, t_2)$ and the smaller of them is called the first period of the signal, see figure 2.2.

[1]Data from a CCD or CMOS chip

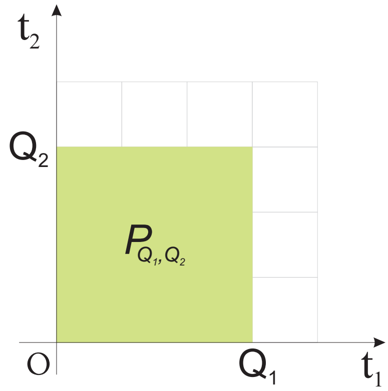

Figure 2.1: Limited continuous signal

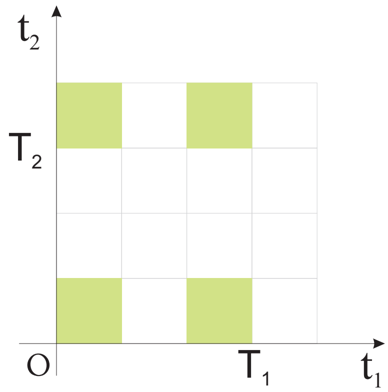

Figure 2.2: Continuous periodic signal

As follows, periodic signal is created by periodization. Periodic continuous signal $\tilde{s}(t_1, t_2)$ with the period (T_1, T_2) can be obtained by the following equation:

$$\tilde{s}(t_1, t_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} s(t_1 + k_1 T_1, t_2 + k_2 T_2). \tag{2.6}$$

2.2 Discrete signal

A discrete signal is a complex function $x(n_1, n_2)$ of two-dimensional variables n_1, n_2, which was created by orthogonal equidistant sampling from the continuous signal $s(t_1, t_2)$ with steps of sampling Δ_1 and Δ_2 such that

$$x(n_1, n_2) = s(n_1 \Delta_1, n_2 \Delta_2), \tag{2.7}$$

where $(n_1, n_2) \in Z^2$, see figure 2.3.

The support of the discrete signal is defined as a subset of Z^2 in which the signal gets non-zero values. When this subset is finite, it is called a discrete signal with finite

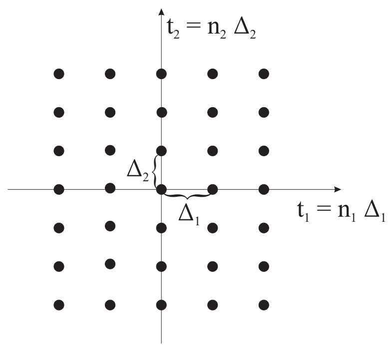

Figure 2.3: Discrete signal

support, see figure 2.4.

$$\begin{array}{ll} x(n_1,n_2) \neq 0 & 0 \leq n_1 \leq M_1-1, \quad 0 \leq n_2 \leq M_2-1, \\ x(n_1,n_2) = 0 & \textit{elsewhere} \end{array} \tag{2.8}$$

The support of this discrete signal is

$$P_{M_1,M_2} = \Big\{(n_1,n_2) \in Z^2 : 0 \leq n_1 \leq M_1-1,\, 0 \leq n_2 \leq M_2-1\Big\}. \tag{2.9}$$

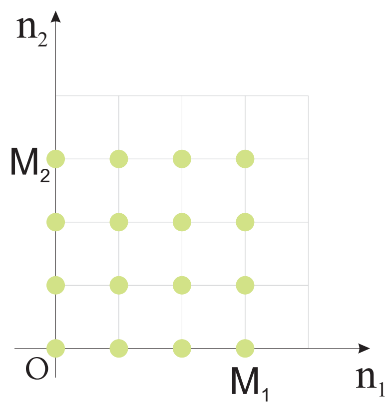

Figure 2.4: Limited discrete signal

Periodic discrete signal is every discrete signal $\tilde{x}(n_1,n_2)$, for which exists couple of integers (N_1,N_2) such that

$$\begin{array}{rl} \tilde{x}(n_1+N_1,n_2) = & \tilde{X}(n_1,n_2), \\ \tilde{x}(n_1,n_2+N_2) = & \tilde{X}(n_1,n_2), \end{array} \tag{2.10}$$

for all $(n_1,n_2) \in Z^2$. The pair (N_1,N_2) is called the periodicity of the signal $\tilde{x}(n_1,n_2)$ and the smaller number of them is called the first period of the signal, see figure 2.2. This discrete periodic signal was obtained from the continuous periodic signal 2.5 (see figure 2.2).

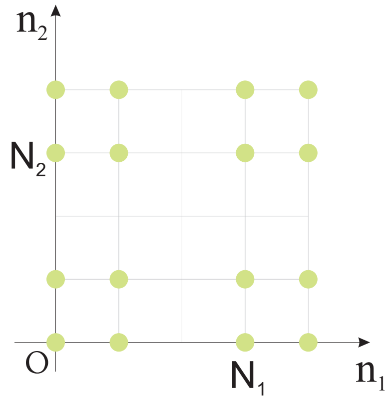

Figure 2.5: Discrete periodic signal

As follows, periodic signal is created by periodization. Periodic discrete signal $\tilde{x}(n_1, n_2)$ with the period (N_1, N_2) can be obtained by this equation

$$\tilde{x}(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(n_1 + k_1 N_1, n_2 + k_2 N_2). \tag{2.11}$$

Consider that 2.6 and 2.11 are convergent.

Chapter 3

Fourier transform and Fourier series of the continuous signal

3.1 Fourier transform of the continuous signal

3.1.1 Definition and conditions of existence

Definition 3.1.1 *Consider a two-dimensional signal* $s(t_1, t_2)$ *with a spectrum of* $S(\Omega_1, \Omega_2)$*.*

$$s(t_1,t_2) = \frac{1}{4\pi^2}\int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} S(\Omega_1,\Omega_2)\, e^{i\Omega_1 t_1 + i\Omega_2 t_2}\, d\Omega_1 d\Omega_2 \tag{3.1}$$

$$S(\Omega_1,\Omega_2) = \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} s(t_1,t_2)\, e^{-i\Omega_1 t_1 - i\Omega_2 t_2}\, dt_1 dt_2 \tag{3.2}$$

Equation 3.1 is called a Inverse Fourier transform of the continuous signal and equation 3.2 is called a Direct Fourier transform of the continuous signal.

These equations describe the relation between the signal $s(t_1, t_2)$ and its spectrum $S(\Omega_1, \Omega_2)$ under certain conditions.

The basic condition is the existence of integrals in equations 3.1 and 3.2. The integral in 3.2 is fulfilled for all Ω_1 and Ω_2 if

$$\int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} |s(t_1,t_2)|\, dt_1 dt_2 < \infty. \tag{3.3}$$

Signals with these properties are called absolutely integrable. This signal is marked as $s \in L_1(R^2)$. With this mark, it is possible to formulate the next theorem.

Theorem 3.1.2 *Consider signal* $s \in L_1(R^2)$*, its spectrum* $S \in L_1(R^2)$ *given by equation 3.2 and also function, which is continuous and represents the signal s. Then the equation 3.1 is useful for all* $(t_1, t_2) \in R^2$*, except zero measure set*[1]*.*

[1] Two dimensional set, for example straight line

Generally is not true, that signal $s \in (R^2)$ has spectrum $S \in (R^2)$. It is necessary to choose the signals s such that

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |s(t_1,t_2)|^2\, dt_1 dt_2 < \infty. \tag{3.4}$$

These signals $s \in L_2(R^2)$ are called square integrable. The existence of the integral 3.3 is not implied by the existence of the integral 3.4. So the equation 3.2 need not make sense for $s \in L_1(R^2)$. The zone of functions where a Fourier transform exists on the whole $L_1(R^2)$ is extended by more general Inverse and Direct Fourier transforms,

$$s_H(t_1,t_2) = \frac{1}{4\pi^2} \lim_{A\to\infty} \iint\limits_{|\Omega_1^2+\Omega_2^2|<A^2} S(\Omega_1,\Omega_2)\, e^{i\Omega_1 t_1 + i\Omega_2 t_2}\, d\Omega_1 d\Omega_2, \tag{3.5}$$

$$S_H(\Omega_1,\Omega_2) = \lim_{A\to\infty} \iint\limits_{|t_1^2+t_2^2|<A^2} s(t_1,t_2)\, e^{-i\Omega_1 t_1 - i\Omega_2 t_2}\, dt_1 dt_2. \tag{3.6}$$

If there exists an integral 3.1 (3.2), then there also exists an integral 3.5 (3.6) and $s_H = s$ ($S_H = S$).

Theorem 3.1.3 *Let $s \in L_2(R^2)$. Then the spectrum S_H, given by the equation 3.6, belongs to $L_2(R^2)$ and*

$$s(t_1,t_2) = \frac{1}{4\pi^2} \lim_{A\to\infty} \iint\limits_{|\Omega_1^2+\Omega_2^2|<A^2} S(\Omega_1,\Omega_2)\, e^{i\Omega_1 t_1 + i\Omega_2 t_2}\, d\Omega_1 d\Omega_2 \tag{3.7}$$

for almost all $(t_1,t_2) \in R^2$.

Consider that if $s \in L_1(R^2)$, then $S_H = S$ and equation 3.7 becomes the equation 3.1.

3.1.2 Properties of the Fourier transform of continuous signal

Consider a signal $s(t_1,t_2)$, its spectrum $S(\Omega_1,\Omega_2)$ and Fourier transform relations (3.1,3.2).

1. Linearity
$$a\ s_1(t_1,t_2) + b\ s_2(t_1,t_2) \Leftrightarrow a\ S_1(\Omega_1,\Omega_2) + b\ S_2(\Omega_1,\Omega_2)$$

2. Rescale
$$s(a\ t_1, b\ t_2) \Leftrightarrow \frac{1}{|ab|}\ S\left(\frac{\Omega_1}{a}, \frac{\Omega_2}{b}\right)$$

3. Signal translation
$$s(t_1 - a, t_2 - b) \Leftrightarrow S(\Omega_1,\Omega_2)\, e^{-ia\Omega_1\ -ib\Omega_2}$$

4. Signal modulation
$$s(t_1,t_2)\, e^{iat_1\ ibt_2} \Leftrightarrow S(\Omega_1\ - a, \Omega_2\ - b)$$

5. Signal differentiation

$$\frac{\partial s(t_1,t_2)}{\partial t_1} \Leftrightarrow i\Omega_1\ S(\Omega_1,\Omega_2)$$
$$\frac{\partial s(t_1,t_2)}{\partial t_2} \Leftrightarrow i\Omega_2\ S(\Omega_1,\Omega_2)$$
$$\frac{\partial^2 s(t_1,t_2)}{\partial t_1\ \partial t_2} \Leftrightarrow -\Omega_1\Omega_2\ S(\Omega_1,\Omega_2)$$

6. Spectrum differentiation

$$\frac{\partial S(\Omega_1,\Omega_2)}{\partial \Omega_1} \Leftrightarrow -it_1\ s(t_1,t_2)$$
$$\frac{\partial S(\Omega_1,\Omega_2)}{\partial \Omega_2} \Leftrightarrow -it_2\ s(t_1,t_2)$$
$$\frac{\partial^2 S(\Omega_1,\Omega_2)}{\partial \Omega_1\ \partial \Omega_2} \Leftrightarrow -it_1t_2\ s(t_1,t_2)$$

7. Transposition

$$s(t_2,t_1) \Leftrightarrow S(\Omega_2,\Omega_1)$$

8. Reflection

$$s(-t_1,t_2) \Leftrightarrow S(-\Omega_1,\Omega_2)$$
$$s(t_1,-t_2) \Leftrightarrow S(\Omega_1,-\Omega_2)$$
$$s(-t_1,-t_2) \Leftrightarrow S(-\Omega_1,-\Omega_2)$$

9. Complex conjugate signal

$$\bar{s}(t_1,t_2) \Leftrightarrow \bar{S}(-\Omega_1,-\Omega_2)$$

10. Real signal

$$s(t_1,t_2) = \bar{s}(t_1,t_2) \Leftrightarrow S(\Omega_1,\Omega_2) = \bar{S}(-\Omega_1, Omega_2)$$

11. Convolution

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s_1(\xi_1,\xi_2)\, s_2(t_1-\xi_1,t_2-\xi_2)\ d\xi_1 d\xi_2 \Leftrightarrow S_1(\Omega_1,\Omega_2)\, S_2(\Omega_1,\Omega_2)$$

12. Signals multiplication

$$s_1(t_1,t_2)\, s_2(t_1,t_2) \Leftrightarrow \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S_1(\varphi_1,\varphi_2)\, S_2(\Omega_1-\varphi_1,\Omega_2-\varphi_2)\, d\varphi_1 d\varphi_2$$

13. Correlation

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s_1(\xi_1,\xi_2)\,\bar{s}_2(t_1+\xi_1,t_2+\xi_2)\;d\xi_1 d\xi_2 \Leftrightarrow S_1(\Omega_1,\Omega_2)\,\bar{S}_2(\Omega_1,\Omega_2)$$

14. Autocorrelation

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(\xi_1,\xi_2)\,\bar{s}(t_1+\xi_1,t_2+\xi_2)\;d\xi_1 d\xi_2 \Leftrightarrow |S(\Omega_1,\Omega_2)|^2$$

15. Parselval's equality

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s_1(t_1,t_2)\,\bar{s}_2(t_1,t_2)\;dt_1 dt_2 = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S_1(\Omega_1,\Omega_2)\,\bar{S}_2(\Omega_1,\Omega_2)\;d\Omega_1 d\Omega_2$$

16. Rayleigh's equality

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |s(t_1,t_2)|^2\;dt_1 dt_2 = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |S(n_1,n_2)|^2\;d\Omega_1 d\Omega_2$$

3.1.3 Amplitude, phase and logarithmic spectrum

Spectrum is represented by a complex function with a real part $Re\,[S(\Omega_1,\Omega_2)]$ and imaginary part $Im\,[S(\Omega_1,\Omega_2)]$. It can be also defined by the amplitude spectrum $A(\Omega_1,\Omega_2)$ and phase spectrum $F(\Omega_1,\Omega_2)$.

$$\begin{aligned} S(\Omega_1,\Omega_2) &= A(\Omega_1,\Omega_2)\,e^{\,i\,F(\Omega_1,\Omega_2)}, & (3.8)\\ A(\Omega_1,\Omega_2) &= |S(\Omega_1,\Omega_2)|, & (3.9)\\ F(\Omega_1,\Omega_2) &= arg\,S(\Omega_1,\Omega_2), & (3.10) \end{aligned}$$

where arg is the main value of argument.

Logarithmic spectrum $L(\Omega_1,\Omega_2)$ is used for better visualization and is equal to logarithm[2] of the amplitude spectrum.

$$L(\Omega_1,\Omega_2) = log\,A(\Omega_1,\Omega_2) = log\,|S(\Omega_1,\Omega_2)| \tag{3.11}$$

3.2 Fourier series of the continuous signal

3.2.1 Definition

It is possible to express a two-dimensional continuous periodic signal $\tilde{s}(t_1,t_2)$ with the period (T_1,T_2) as a sum of weighted exponentials.

$$\tilde{s}(t_1,t_2) = \sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} S(n_1,n_2)\,e^{\left(2\pi i\left(\frac{n_1 t_1}{T_1}+\frac{n_2 t_2}{T_2}\right)\right)} \tag{3.12}$$

[2]Natural logarithm

with the Fourier coefficients $S(n_1, n_2)$ where

$$S(n_1, n_2) = \frac{1}{T_1 T_2} \int\limits_{a}^{a+T_1} \int\limits_{b}^{b+T_2} \tilde{s}(t_1, t_2)\, e^{\left(-2\pi i \left(\frac{n_1 t_1}{T_1} + \frac{n_2 t_2}{T_2}\right)\right)} dt_1 dt_2 \tag{3.13}$$

where a, b are real numbers. It is advantageous to choose $a = \frac{-T_1}{2}$ and $b = \frac{-T_2}{2}$.

$$S(n_1, n_2) = \frac{1}{T_1 T_2} \int\limits_{\frac{-T_1}{2}}^{\frac{T_1}{2}} \int\limits_{\frac{-T_2}{2}}^{\frac{T_2}{2}} \tilde{s}(t_1, t_2)\, e^{\left(-2\pi i \left(\frac{n_1 t_1}{T_1} + \frac{n_2 t_2}{T_2}\right)\right)} dt_1 dt_2 \tag{3.14}$$

3.2.2 Convergence of the Fourier series

The Fourier series of the periodic signal $\tilde{s}(t_1, t_2)$ converges to this signal if

$$\int\limits_{\frac{-T_1}{2}}^{\frac{T_1}{2}} \int\limits_{\frac{-T_2}{2}}^{\frac{T_2}{2}} |\tilde{s}(t_1, t_2)|^2 dt_1 dt_2 \ < \ \infty \tag{3.15}$$

and if

$$\lim_{N \to \infty} \int\limits_{\frac{-T_1}{2}}^{\frac{T_1}{2}} \int\limits_{\frac{-T_2}{2}}^{\frac{T_2}{2}} \left| \tilde{s}(t_1, t_2) - \sum_{n_1=-N}^{N} \sum_{n_2=-N}^{N} S(n_1, n_2)\, e^{\left(2\pi i \left(\frac{n_1 t_1}{T_1} + \frac{n_2 t_2}{T_2}\right)\right)} \right|^2 dt_1 dt_2 \ = \ 0. \tag{3.16}$$

Periodic signals $\tilde{s}$ with the property 3.15 are signed as $\tilde{s} \in L_2(I_{T_1,T_2})$.

Theorem 3.2.1 *Select complex numbers $S(n_1, n_2)$ where n_1, n_2 are integers such that*

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |S(n_1, n_2)|^2 \ < \ \infty. \tag{3.17}$$

Then the series

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} S(n_1, n_2)\, e^{\left(2\pi i \left(\frac{n_1 t_1}{T_1} + \frac{n_2 t_2}{T_2}\right)\right)} dt_1 dt_2 \tag{3.18}$$

converges to the signal $\tilde{s}(t_1, t_2) \in L_2(I_{T_1,T_2})$ in the sense of the limit 3.16.

Theorem 3.2.2 *Let signal $s(t_1, t_2)$ have, in its interval $I_{T_1,T_2} \equiv \langle -\frac{T_1}{2}, \frac{T_1}{2} \rangle \times \langle -\frac{T_2}{2}, \frac{T_2}{2} \rangle$, continuous derivations $\frac{\partial \tilde{s}}{\partial t_1}$, $\frac{\partial \tilde{s}}{\partial t_2}$ and $\frac{\partial^2 s}{\partial t_1 \partial t_2}$. Then 3.12 is true in all points in the interval (t_1, t_2), where coefficients $S(n_1, n_2)$ are given by the equation 3.14.*

If we consider series with coefficients $S(n_1, n_2)$, hence

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |S(n_1, n_2)| \ < \ \infty, \tag{3.19}$$

converges absolutely, then the series converges uniformly. So the signal $\tilde{s}(t_1, t_2)$ exists and is continuous.

3.2.3 Properties of the Fourier series

Consider signal $\tilde{s}(t_1, t_2)$ is integrable in the rectangle $\langle -\frac{T_1}{2}, \frac{T_1}{2} \rangle \times \langle -\frac{T_2}{2}, \frac{T_2}{2} \rangle$ and its Fourier series with coefficients $S(n_1, n_2)$ converges. So for the transformation relations 3.12 and 3.14, the following properties are true.

1. Linearity
$$a\ \tilde{s}_1(t_1, t_2) + b\ \tilde{s}_2(t_1, t_2) \Leftrightarrow a\ S_1(n_1, n_2) + b\ S_2(n_1, n_2)$$

2. Rescale
$$\tilde{s}(a\ t_1, b\ t_2) \Leftrightarrow S(n_1, n_2)$$

3. Signal translation
$$\tilde{s}(t_1 - a, t_2 - b) \Leftrightarrow S(n_1, n_2)\, e^{\left(-2\pi i \left(\frac{a n_1}{T_1} + \frac{b n_2}{T_2}\right)\right)}$$

4. Signal modulation
$$\tilde{s}(t_1, t_2)\, e^{\left(2\pi i \left(\frac{m_1 t_1}{T_1} + \frac{m_2 t_2}{T_2}\right)\right)} \Leftrightarrow S(n_1 - m_1, n_2 - m_2)$$

5. Signal differentiation
$$\begin{aligned}
\frac{\partial \tilde{s}(t_1, t_2)}{\partial t_1} &\quad \Leftrightarrow \quad 2\pi i\, \frac{n_1}{T_1}\, S(n_1, n_2) \\
\frac{\partial \tilde{s}(t_1, t_2)}{\partial t_2} &\quad \Leftrightarrow \quad 2\pi i\, \frac{n_2}{T_2}\, S(n_1, n_2) \\
\frac{\partial^2 \tilde{s}(t_1, t_2)}{\partial t_1\ \partial t_2} &\quad \Leftrightarrow \quad -4\pi^2\, \frac{n_1 n_2}{T_1 T_2}\, S(n_1, n_2)
\end{aligned}$$

6. Transposition
$$\tilde{s}(t_2, t_1) \Leftrightarrow S(n_2, n_1)$$

7. Reflection
$$\begin{aligned}
\tilde{s}(-t_1, t_2) &\quad \Leftrightarrow \quad S(-n_1, n_2) \\
\tilde{s}(t_1, -t_2) &\quad \Leftrightarrow \quad S(n_1, -n_2) \\
\tilde{s}(-t_1, -t_2) &\quad \Leftrightarrow \quad S(-n_1, -n_2)
\end{aligned}$$

8. Complex conjugate signal
$$\bar{\tilde{s}}(t_1, t_2) \Leftrightarrow \bar{\tilde{s}}(-n_1, -n_2)$$

9. Real signal
$$\tilde{s}(t_1, t_2) = \bar{\tilde{s}}(t_1, t_2) \Leftrightarrow S(n_1, n_2) = \bar{S}(-n_1, -n_2)$$

10. Periodic convolution

$$\frac{1}{T_1T_2}\int\limits_{\frac{-T_1}{2}}^{\frac{T_1}{2}}\int\limits_{\frac{-T_2}{2}}^{\frac{T_2}{2}} \tilde{s}_1(\xi_1,\xi_2)\,\tilde{s}_2(t_1-\xi_1,t_2-\xi_2)\ d\xi_1 d\xi_2 \Leftrightarrow S_1(n_1,n_2)\,S_2(n_1,n_2)$$

11. Signals multiplication

$$\tilde{s}_1(t_1,t_2)\,\tilde{s}_2(t_1,t_2) \Leftrightarrow \sum_{m_1=-\infty}^{\infty}\sum_{m_2=-\infty}^{\infty} S_1(m_1,m_2)\,S_2(n_1-m_1,n_2-m_2)$$

12. Periodic correlation

$$\frac{1}{T_1T_2}\int\limits_{\frac{-T_1}{2}}^{\frac{T_1}{2}}\int\limits_{\frac{-T_2}{2}}^{\frac{T_2}{2}} \tilde{s}_1(\xi_1,\xi_2)\,\bar{\tilde{s}}_2(t_1+\xi_1,t_2+\xi_2)\ d\xi_1 d\xi_2 \Leftrightarrow S_1(n_1,n_2)\,\bar{S}_2(n_1,n_2)$$

13. Periodic autocorrelation

$$\frac{1}{T_1T_2}\int\limits_{\frac{-T_1}{2}}^{\frac{T_1}{2}}\int\limits_{\frac{-T_2}{2}}^{\frac{T_2}{2}} \tilde{s}(\xi_1,\xi_2)\,\bar{\tilde{s}}(t_1+\xi_1,t_2+\xi_2)\ d\xi_1 d\xi_2 \Leftrightarrow |S(n_1,n_2)|^2$$

14. Parselval's equality

$$\frac{1}{T_1T_2}\int\limits_{\frac{-T_1}{2}}^{\frac{T_1}{2}}\int\limits_{\frac{-T_2}{2}}^{\frac{T_2}{2}} \tilde{s}_1(t_1,t_2)\,\bar{\tilde{s}}_2(t_1,t_2)\ dt_1 dt_2 = \sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} S_1(n_1,n_2)\,\bar{S}_2(n_1,n_2)$$

15. Rayleigh's equality

$$\frac{1}{T_1T_2}\int\limits_{\frac{-T_1}{2}}^{\frac{T_1}{2}}\int\limits_{\frac{-T_2}{2}}^{\frac{T_2}{2}} |\tilde{s}(t_1,t_2)|^2\ dt_1 dt_2 = \sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} |S_1(n_1,n_2)|^2$$

3.2.4 Amplitude, phase and logarithmic spectrum

Fourier coefficients $S(n_1,n_2)$ of the continuous signal $\tilde{s}(t_1,s_2)$ constitute a discrete spectrum. It can also be declared by the amplitude spectrum $A(n_1,n_2)$ and phase spectrum $F(n_1,n_2)$.

$$\begin{aligned} S(n_1,n2) &= A(n_1,n_2)\,e^{i\,F(n_1,n_2)}, && (3.20)\\ A(n_1,n_2) &= |S(n_1,n_2)|, && (3.21)\\ F(n_1,n_2) &= arg\,S(n_1,n_2), && (3.22) \end{aligned}$$

where arg is the main value of the argument.

The logarithmic spectrum $L(n_1,n_2)$ is used for better visualization and is equal to logarithm[3] of the amplitude spectrum.

$$L(n_1,n_2) = log\,A(n_1,n_2) = log\,|S(n_1,n_2)| \qquad (3.23)$$

[3] Natural logarithm

3.2.5 Fourier series for the continuous signal in finite interval

Consider the signal $s(t_1, t_2)$ in the interval $I_{T_1,T_2} \equiv \langle -\frac{T_1}{2}, \frac{T_1}{2}\rangle \times \langle -\frac{T_2}{2}, \frac{T_2}{2}\rangle$. Equations 3.12 and 3.14 are true also for the non-periodic signal if it satisfies periodic boundary conditions

$$\begin{aligned} s\Big(t_1, -\frac{T_2}{2}\Big) &= s(t_1, \frac{T_2}{2}) \qquad for \ \Big\langle t_1 \in -\frac{T_1}{2}, \frac{T_1}{2}\Big\rangle, \\ s\Big(-\frac{T_1}{2}, t_2\Big) &= s(\frac{T_1}{2}, t_2) \qquad for \ \Big\langle t_2 \in -\frac{T_2}{2}, \frac{T_2}{2}\Big\rangle. \end{aligned}$$

3.3 Relation between Fourier series and Fourier Transform

S_T denotes the Fourier transform 3.2 and $S(n_1, n_2)$ are Fourier coefficients 3.14 for the Fourier series. If $s(t_1, t_2) = 0$ outside of the interval $\langle a, a + T_1\rangle \times \langle b, b + T_1\rangle$, then

$$S_T\,(2\pi\,\frac{n_1}{T_1}, 2\pi\,\frac{n_2}{T_2}) = T_1 T_2\, S(n_1 n_2) \tag{3.24}$$

Using the discretization of the spectrum is possible to obtain Fourier series from the Fourier transform. Suppose that for the continuous signal there exists a Fourier transform and calculate the values in points $(n_1\Delta_1, n_2\Delta_2)$.

$$S_T(n_1\Delta_1, n_2\Delta_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(t_1, t_2)\, e^{-in_1\Delta_1 t_1 - in_2\Delta_2 t_2}\, dt_1 dt_2 \tag{3.25}$$

where $\Delta_1 = \frac{2\pi}{T_1}$ and $\Delta_2 = \frac{2\pi}{T_2}$. So the equation becomes

$$S_T(2\pi\,\frac{n_1}{T_1}, 2\pi\,\frac{n_2}{T_2}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(t_1, t_2)\, e^{-in_1\frac{2\pi}{T_1}t_1 - in_2\frac{2\pi}{T_2}t_2}\, dt_1 dt_2 \tag{3.26}$$

The integration domain is divided into two rectangles $Q(l_1, l_2) \equiv \langle l_1T_1 - \frac{T_1}{2}, l_1T_1 + \frac{T_1}{2}\rangle \times \langle l_2T_2 - \frac{T_2}{2}, l_2T_2 + \frac{T_2}{2}\rangle$. So

$$S_T(2\pi\,\frac{n_1}{T_1}, 2\pi\,\frac{n_2}{T_2}) = \sum_{l_1=-\infty}^{\infty}\sum_{l_2=-\infty}^{\infty}\iint_{Q(l_1,l_2)} s(t_1, t_2)\, e^{-in_1\frac{2\pi}{T_1}t_1 - in_2\frac{2\pi}{T_2}t_2}\, dt_1 dt_2 \tag{3.27}$$

Using substitutions $\xi_1 = t_1 - l_1T_1$, $\xi_2 = t_2 - l_2T_2$ and considering that series

$$\sum_{l_1=-\infty}^{\infty}\sum_{l_2=-\infty}^{\infty} s(\xi_1 + l_1T_1, \xi_2 + l_2T_2) \tag{3.28}$$

in the rectangle $\langle -\frac{T_1}{2}, \frac{T_1}{2}\rangle \times \langle -\frac{T_2}{2}, \frac{T_2}{2}\rangle$ is uniformly convergent, we obtain

$$S_T(2\pi\,\frac{n_1}{T_1}, 2\pi\,\frac{n_2}{T_2}) = \int_{-\frac{T_1}{2}}^{\frac{T_1}{2}}\int_{-\frac{T_2}{2}}^{\frac{T_2}{2}} \sum_{l_1=-\infty}^{\infty}\sum_{l_2=-\infty}^{\infty} s(\xi_1 + l_1T_1, \xi_2 + l_2T_2)\, e^{-in_1\frac{2\pi\xi_1}{T_1} - in_2\frac{2\pi\xi_2}{T_2}}\, d\xi_1 d\xi_2. \tag{3.29}$$

By chosing

$$\tilde{s}(\xi_1,\xi_2) = \sum_{l_1=-\infty}^{\infty}\sum_{l_2=-\infty}^{\infty} s(\xi_1+l_1T_1,\xi_2+l_2T_2) \tag{3.30}$$

$$S(n_1,n_2) = \frac{1}{T_1T_2}\,S_T(2\pi\,\frac{n_1}{T_1},2\pi\,\frac{n_2}{T_2}). \tag{3.31}$$

we obtain equation

$$S(n_1,n_2) = \frac{1}{T_1T_2}\int_{\frac{-T_1}{2}}^{\frac{T_1}{2}}\int_{\frac{-T_2}{2}}^{\frac{T_2}{2}} \tilde{s}(\xi_1,\xi_2)\,e^{\left(-2\pi i\left(\frac{n_1\xi_1}{T_1}+\frac{n_2\xi_2}{T_2}\right)\right)}d\xi_1 d\xi_2 \tag{3.32}$$

which is identical to 3.14.

Chapter 4

Fourier transform and Fourier series of the discrete signal

4.1 Fourier transform of the discrete signal

4.1.1 Definition and conditions of existence

Definition 4.1.1 *Consider a two-dimensional signal $x(t_1, t_2)$ and $\tilde{X}(\omega_1, \omega_2)$ is its spectrum.*

$$x(n_1, n_2) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \tilde{X}(\omega_1, \omega_2)\, e^{i\omega_1 n_1 + i\omega_2 n_2}\, d\omega_1 d\omega_2 \tag{4.1}$$

$$\tilde{X}(\omega_1, \omega_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2-\infty}^{\infty} x(n_1, n_2)\, e^{-i\omega_1 n_1 - i\omega_2 n_2} \tag{4.2}$$

The equation 4.1 is called an Inverse Fourier transform of discrete signal and the equation 4.2 is called a Direct Fourier transform of discrete signal.

A basic condition is the existence of integrals in the equation 4.2 and also that the series in the equation 4.1 converges uniformly to the interval $I_{2\pi,2\pi} \equiv \langle -\pi, \pi \rangle \times \langle -\pi, \pi \rangle$. From the analogy to the Fourier series of continuous signal, it is clear that the Fourier transform $\tilde{X} \in L_2(I_{2\pi,2\pi})$ exists if its series converges absolutely.

4.1.2 Properties of the Fourier transform of the discrete signal

Consider signal $x(n_1, n_2)$ and its spectrum $\tilde{X}(\omega_1, \omega_2)$. The Fourier transform (4.1,4.2) have the following properties:

1. Linearity

$$a\; x_1(n_1, n_2) + b\; x_2(n_1, n_2) \Leftrightarrow a\; \tilde{X}_1(\omega_1, \omega_2) + b\; \tilde{X}_2(\omega_1, \omega_2)$$

2. Signal translation

$$x(n_1 - m_1, n_2 - m_2) \Leftrightarrow \tilde{X}(\omega_1, \omega_2)\, e^{-im_1\omega_1 \; -im_2\omega_2}$$

3. Signal modulation

$$x(n_1, n_2)\, e^{i\varphi_1 n_1 + i\varphi_2 n_2} \Leftrightarrow \tilde{X}(\omega_1 \; - \varphi_1, \omega_2 \; - \varphi_2)$$

4. Spectrum differentiation

$$\begin{aligned}
\frac{\partial \tilde{X}(\omega_1, \omega_2)}{\partial \omega_1} \;&\Leftrightarrow\; -in_1 \; x(n_1, n_2) \\
\frac{\partial \tilde{X}(\omega_1, \omega_2)}{\partial \omega_2} \;&\Leftrightarrow\; -in_2 \; x(n_1, n_2) \\
\frac{\partial^2 \tilde{X}(\omega_1, \omega_2)}{\partial \omega_1 \; \partial \omega_2} \;&\Leftrightarrow\; -n_1 n_2 \; x(n_1, n_2)
\end{aligned}$$

5. Transposition

$$x(n_2, n_1) \Leftrightarrow \tilde{X}(\omega_2, \omega_1)$$

6. Reflection

$$\begin{aligned}
x(-n_1, n_2) \;&\Leftrightarrow\; \tilde{X}(-\omega_1, \omega_2) \\
x(n_1, -n_2) \;&\Leftrightarrow\; \tilde{X}(\omega_1, -\omega_2) \\
x(-n_1, -n_2) \;&\Leftrightarrow\; \tilde{X}(-\omega_1, -\omega_2)
\end{aligned}$$

7. Complex conjugate signal

$$\bar{x}(n_1, n_2) \Leftrightarrow \bar{X}(-\omega_1, -\omega_2)$$

8. Real signal

$$x(n_1, n_2) = \bar{x}(n_1, n_2) \Leftrightarrow \tilde{X}(\omega_1, \omega_2) = \bar{\tilde{X}}(-\omega_1, -\omega_2)$$

9. Discrete convolution

$$\sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} x_1(m_1, m_2)\, x_2(n_1 - m_1, n_2 - m_2) \Leftrightarrow \tilde{X}_1(\omega_1, \omega_2)\, \tilde{X}_2(\omega_1, \omega_2)$$

10. Signals multiplication

$$x_1(n_1, n_2)\, x_2(n_1, n_2) \Leftrightarrow \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \tilde{X}_1(\varphi_1, \varphi_2)\, \tilde{X}_2(\omega_1 - \varphi_1, \omega_2 - \varphi_2)\, d\varphi_1 d\varphi_2$$

11. Discrete correlation

$$\sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} x(m_1, m_2)\, \bar{x}_2(m_1 + n_1, m_2 + n_2) \Leftrightarrow \tilde{X}_1(\omega_1, \omega_2)\, \bar{\tilde{X}}_2(\omega_1, \omega_2)$$

12. Discrete autocorrelation

$$\sum_{m_1=-\infty}^{\infty}\sum_{m_2=-\infty}^{\infty} x(m_1,m_2)\,\bar{x}(m_1+n_1,m_2+n_2) \Leftrightarrow |\tilde{X}(\omega_1,\omega_2)|^2$$

13. Parselval's equality

$$\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} x_1(n_1,n_2)\,\bar{x}_2(n_1,n_2) = \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \tilde{X}_1(\omega_1,\omega_2)\,\bar{\tilde{X}}_2(\omega_1,\omega_2)\ d\omega_1 d\omega_2$$

14. Rayleigh's equality

$$\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} |x(n_1,n_2)|^2 = \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |\tilde{X}(\omega_1,\omega_2)|^2\ d\omega_1 d\omega_2$$

4.1.3 Amplitude, phase and logarithmic spectrum

The discrete spectrum is represented by complex function with the real $Re\,[\tilde{X}(\omega_1,\omega_2)]$ and imaginary part $Im\,[\tilde{X}(\Omega_1,\Omega_2)]$. It can be also defined by the the amplitude spectrum $\tilde{A}(\omega_1,\omega_2)$ and phase spectrum $\tilde{F}(\omega_1,\omega_2)$.

$$\begin{aligned}\tilde{X}(\omega_1,\omega_2) &= \tilde{A}(\omega_1,\omega_2)\,e^{i\,\tilde{F}(\omega_1,\omega_2)}, && (4.3)\\ \tilde{A}(\omega_1,\omega_2) &= |\tilde{S}(\Omega_1,\Omega_2)|, && (4.4)\\ \tilde{F}(\omega_1,\omega_2) &= arg\,\tilde{X}(\omega_1,\omega_2), && (4.5)\end{aligned}$$

where arg is the main value of the argument.

The logarithmic spectrum $\tilde{L}(\omega_1,\omega_2)$ is used for better visualization and is equal to the logarithm[1] of the amplitude spectrum.

$$\tilde{L}(\omega_1,\omega_2) = log\,\tilde{A}(\omega_1,\omega_2) = log\,|\tilde{S}(\omega_1,\omega_2)| \tag{4.6}$$

4.2 Fourier series of the discrete signal

Definition 4.2.1 *Let 4.7 be the equation for the computation of Fourier series coefficients* $\tilde{X}(l_1,l_2)$.

$$\tilde{X}(l_1,l_2) \;=\; \frac{1}{N_1N_2}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} \tilde{x}(n_1,n_2)\,e^{\left(-2\pi i\left(\frac{n_1l_1}{N_1}+\frac{n_2l_2}{N_2}\right)\right)} \tag{4.7}$$

then the expression 4.8

$$\tilde{x}(n_1,n_2) = \sum_{l_1=0}^{N_1-1}\sum_{l_2=0}^{N_1-1} \tilde{X}(l_1,l_2)\,e^{\left(2\pi i\left(\frac{n_1l_1}{N_1}+\frac{n_2l_2}{2}\right)\right)} \tag{4.8}$$

is called the Fourier series of a discrete signal.

[1] Natural logarithm

4.2.1 Properties of the Fourier series

Consider $\tilde{x}(n_1, n_2) \Leftrightarrow \tilde{X}(l_1, l_2)$ and the transformations 4.7 and 4.8. Then the Fourier series has the following properties:

1. Linearity
$$a\ \tilde{x}_1(n_1, n_2) + b\ \tilde{x}_2(n_1, n_2) \Leftrightarrow a\ \tilde{X}_1(l_1, l_2) + b\ \tilde{X}_2(l_1, l_2)$$

2. Signal translation
$$\tilde{x}(n_1 - m_1, n_2 - m_2) \Leftrightarrow \tilde{X}(l_1, l_2)\, e^{\left(-2\pi i\left(\frac{m_1 l_1}{N_1} + \frac{m_2 l_2}{N_2}\right)\right)}$$

3. Signal modulation
$$\tilde{x}(n_1, n_2)\, e^{\left(2\pi i\left(\frac{m_1 n_1}{N_1} + \frac{m_2 n_2}{N_2}\right)\right)} \Leftrightarrow \tilde{X}(l_1 - m_1, l_2 - m_2)$$

4. Transposition
$$\tilde{x}(n_2, n_1) \Leftrightarrow \tilde{X}(l_2, l_1)$$

5. Reflection
$$\begin{aligned} \tilde{x}(-n_1, n_2) &\Leftrightarrow \tilde{X}(-l_1, l_2) \\ \tilde{x}(n_1, -n_2) &\Leftrightarrow \tilde{X}(l_1, -l_2) \\ \tilde{x}(-n_1, -n_2) &\Leftrightarrow \tilde{X}(-l_1, -l_2) \end{aligned}$$

6. Complex conjugate signal
$$\bar{\tilde{x}}(n_1, n_2) \Leftrightarrow \bar{\tilde{X}}(-l_1, -l_2)$$

7. Real signal
$$\tilde{x}(n_1, n_2) = \bar{\tilde{x}}(n_1, n_2) \Leftrightarrow \tilde{X}(l_1, l_2) = \bar{\tilde{X}}(-l_1, -l_2)$$

8. Discrete periodic convolution
$$\frac{1}{N_1 N_2} \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} \tilde{x}_1(m_1, m_2)\, \tilde{x}_2(n_1 - m_1, n_2 - m_2) \Leftrightarrow \tilde{X}_1(l_1, l_2)\, \tilde{X}_2(l_1, l_2)$$

9. Signals multiplication
$$\tilde{x}_1(n_1, n_2)\, \tilde{x}_2(n_1, n_2) \Leftrightarrow \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} \tilde{X}_1(m_1, m_2)\, \tilde{X}_2(l_1 - m_1, l_2 - m_2)$$

10. Discrete periodic correlation
$$\frac{1}{N_1 N_2} \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} \tilde{x}_1(m_1, m_2)\, \bar{\tilde{x}}_2(n_1 + m_1, n_2 + m_2) \Leftrightarrow \tilde{X}_1(l_1, l_2)\, \bar{\tilde{X}}_2(l_1, l_2)$$

11. Discrete periodic autocorrelation

$$\frac{1}{N_1N_2}\sum_{m_1=0}^{N_1-1}\sum_{m_2=0}^{N_2-1}\tilde{x}(m_1,m_2)\,\bar{\tilde{x}}(n_1+m_1,n_2+m_2)\Leftrightarrow|\tilde{X}(l_1,l_2)|^2$$

12. Parselval's equality

$$\frac{1}{N_1N_2}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}\tilde{x}_1(n_1,n_2)\,\bar{\tilde{x}}_2(n_1,n_2)=\sum_{l_1=0}^{N_1-1}\sum_{l_2=0}^{N_2-1}\tilde{X}_1(l_1,l_2)\,\bar{\tilde{X}}_2(l_1,l_2)$$

13. Rayleigh's equality

$$\frac{1}{N_1N_2}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}|\tilde{x}(n_1,n_2)|^2=\sum_{l_1=0}^{N_1-1}\sum_{l_2=0}^{N_2-1}|\tilde{X}(l_1,l_2)|^2$$

4.2.2 Amplitude, phase and logarithmic spectrum

The amplitude spectrum $\tilde{A}(l_1,l_2)$ and phase spectrum $\tilde{F}(l_1,l_2)$ are defined by these equations:

$$\begin{aligned}\tilde{X}(l_1,l_2) &= \tilde{A}(l_1,l_2)\,e^{\,i\,\tilde{F}(l_1,l_2)}, &(4.9)\\ \tilde{A}(l_1,l_2) &= |\tilde{X}(l_1,l_2)|, &(4.10)\\ \tilde{F}(l_1,l_2) &= arg\,\tilde{X}(l_1,l_2), &(4.11)\end{aligned}$$

where arg is the main value of the argument.

The logarithmic spectrum $\tilde{L}(l_1,l_2)$ is used for better visualization and is equal to the logarithm[2] of the amplitude spectrum.

$$\tilde{L}(l_1,l_2)=log\,\tilde{A}(l_1,l_2)=log\,|\tilde{X}(n_1,n_2)| \tag{4.12}$$

4.3 Relation between Fourier series and Fourier Transform

$\tilde{X}_T(\omega_1,\omega_2)$ denotes the Fourier transform of the discrete signal $x(n_1,n_2)$.

$$\tilde{X}_T(\omega_1,\omega_2)=\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}x(n_1,n_2)\,e^{-i\omega_1n_1-i\omega_2n_2}. \tag{4.13}$$

Let $\omega_1=\frac{2\pi l_1}{N_1}$, $\omega_2=\frac{2\pi l_2}{N_2}$, $n_1=m_1+k_1N_1$ and $n_2=m_2+k_2N_2$ so

$$\tilde{X_T}\Big(\frac{2\pi l_1}{N_1},\frac{2\pi l_2}{N_2}\Big)=\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}\sum_{m_1=0}^{N_1-1}\sum_{m_2=0}^{N_2-1}x(m_1+k_1N_1,m_2+k_2N_2)\,e^{\left(-2\pi i\left(\frac{l_1m_1}{N_1}+\frac{l_2m_2}{N_2}\right)\right)} \tag{4.14}$$

[2]Natural logarithm

If this series converges absolutely (4.15), it is possible to reorder the series 4.14.

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |x(n_1, n_2)| \; < \; \infty \tag{4.15}$$

And if

$$\tilde{x}(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(m_1 + k_1 N_1, m_2 + k_2 N_2),$$

$$\tilde{X}(l_1, l_2) = \frac{1}{N_1 N_2} \; \tilde{X}_T(2\pi \frac{l_1}{N_1}, 2\pi \frac{l_2}{N_2}),$$

then the equation

$$\tilde{X}(l_1, l_2) = \frac{1}{N_1 N_2} \, \tilde{x}_T(m_1, m_2) \, e^{\left(-2\pi i \left(\frac{l_1 m_1}{N_1} + \frac{l_2 m_2}{N_2}\right)\right)} \tag{4.16}$$

gives the values of the Fourier transform $\tilde{X}_T(2\pi \frac{l_1}{N_1}, 2\pi \frac{l_2}{N_2})$ for the non-perodical signal x.

Chapter 5

2-D discrete Fourier transform

5.1 Definition

Consider that $y(n_1, n_2)$ is a signal and $Y(n_1, n_2)$ is its spectrum, then the 2-D Fourier transform (DFT) is defined by these two equations:

$$Y(l_1, l_2) = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} y(n_1, n_2)\, e^{-2\pi i \left(\frac{n_1 l_1}{N_1} + \frac{n_2 l_2}{N_2}\right)} \tag{5.1}$$

$$y(n_1, n_2) = \sum_{l_1=0}^{N_1-1} \sum_{l_2=0}^{N_2-1} Y(l_1, l_2)\, e^{2\pi i \left(\frac{n_1 l_1}{N_1} + \frac{n_2 l_2}{N_2}\right)} \tag{5.2}$$

for all $0 \le n_1 \le N_1 - 1$, $0 \le n_2 \le N_2 - 1$. Then the equation 5.1 is called a Direct 2-D Fourier transform and the equation 5.2 is called Inverse 2-D Fourier transform.

5.2 Properties of DFT

The properties of DFT are the same as the properties of the Fourier series of discrete signal.

1. Linearity

$$a\, y_1(n_1, n_2) + b\, y_2(n_1, n_2) \Leftrightarrow a\, Y_1(l_1, l_2) + b\, Y_2(l_1, l_2)$$

2. Cyclic translation

$$y((n_1 - m_1)\, mod\, N_1, (n_2 - m_2)\, mod\, N_2) \Leftrightarrow Y(l_1, l_2)\, e^{\left(-2\pi i \left(\frac{m_1 l_1}{N_1} + \frac{m_2 l_2}{N_2}\right)\right)}$$

3. Modulation

$$y(n_1, n_2)\, e^{\left(2\pi i \left(\frac{m_1 n_1}{N_1} + \frac{m_2 n_2}{N_2}\right)\right)} \Leftrightarrow Y((l_1 - m_1)\, mod\, N_1, (l_2 - m_2)\, mod\, N_2)$$

4. Transposition

$$y(n_2, n_1) \Leftrightarrow Y(l_2, l_1)$$

5. Reflection

$$\begin{aligned} y((N_1-n_1)\ mod\, N_1, n_2) &\Leftrightarrow Y((N_1-l_1)\ mod\, N_1, l_2) \\ y(n_1,(N_2-n_2)\ mod\, N_2) &\Leftrightarrow Y(l_1,(N_2-l_2)\ mod\, N_2) \\ y((N_1-n_1)\ mod\, N_1,(N_2-n_2)\ mod\, N_2) &\Leftrightarrow Y((N_1-l_1)\ mod\, N_1,(N_2-l_2)\ mod\, N_2) \end{aligned}$$

6. Complex conjugate values

$$\bar{y}(n_1,n_2) \Leftrightarrow \bar{Y}((N_1-l_1)\ mod\, N_1,(N_2-l_2)\ mod\, N_2)$$

7. Real values

$$y(n_1,n_2)=\bar{y}(n_1,n_2) \Leftrightarrow Y(l_1,l_2)=\bar{Y}((N_1-l_1)\ mod\, N_1,(N_2-l_2)\ mod\, N_2)$$

8. Discrete periodic convolution

$$\frac{1}{N_1N_2}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} y_1(m_1,m_2)\, y_2((n_1-m_1)\ mod\, N_1,(n_2-m_2)\ mod\, N_2) \Leftrightarrow Y_1(l_1,l_2)\, Y_2(l_1,l_2)$$

9. Multiplication

$$y_1(n_1,n_2)\, y_2(n_1,n_2) \Leftrightarrow \sum_{m_1=0}^{N_1-1}\sum_{m_2=0}^{N_2-1} Y_1(m_1,m_2)\, Y_2((l_1-m_1)\ mod\, N_1,(l_2-m_2)\ mod\, N_2)$$

10. Discrete periodic correlation

$$\frac{1}{N_1N_2}\sum_{m_1=0}^{N_1-1}\sum_{m_2=0}^{N_2-1} (m_1,m_2)\, \bar{y}((n_1+m_1)\ mod\, N_1,(n_2+m_2)\ mod\, N_2) \Leftrightarrow Y_1(l_1,l_2)\, \bar{Y}_2(l_1,l_2)$$

11. Discrete periodic autocorrelation

$$\frac{1}{N_1N_2}\sum_{m_1=0}^{N_1-1}\sum_{m_2=0}^{N_2-1} y(m_1,m_2)\, \bar{y}((n_1+m_1)\ mod\, N_1,(n_2+m_2)\ mod\, N_2) \Leftrightarrow |Y(l_1,l_2)|^2$$

12. Parselval's equality

$$\frac{1}{N_1N_2}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} y_1(n_1,n_2)\, \bar{y}_2(n_1,n_2) = \sum_{l_1=0}^{N_1-1}\sum_{l_2=0}^{N_2-1} Y_1(l_1,l_2)\, \bar{Y}_2(l_1,l_2)$$

13. Rayleigh's equality

$$\frac{1}{N_1N_2}\sum_{m_1=0}^{N_1-1}\sum_{m_2=0}^{N_2-1} |y(n_1,n_2)|^2 = \sum_{l_1=0}^{N_1-1}\sum_{l_2=0}^{N_2-1} |Y(l_1,l_2)|^2$$

Chapter 6

Program for the analysis of the nozzle

6.1 Introduction

In the Fluid flow and heat transfer laboratory of the University of technology, two experiments took place. The first was the pulsation effect on the intensity of the heat transfer and the second was the pulsation effect on the disposal of secondary scales. After these experiments, we took pictures of the nozzles for pressures 50 and 200 bar during the night. For these pressures, water elements can reach velocity $300\,ms^{-1}$. So it was necessary to brighten the track of the nozzle by a light sheet of pulsed laser. We used the mathematical method to characterise the jet. The program was made in Delphi 7.

6.2 Program face

On the right side of the form is a Memo list where you can get information about the picture and its processing.

On the top there is menu divided to files, tools, view and others. The file menu contains the procedure for opening images, saving an image and closing the program. In the second menu there are three procedures, namely Transformation procedure, Integration procedure, Column sum procedure and Image evaluation. Transformation will do the Fourier transform and display the permuted amplitude and logarithm spectrum. The second procedure, Integration, will show the representation of the selected frequency, which represents the determinate drop. To be more precise, it is not a drop, but a more complicated formation. The third procedure serves to check the lengthwise characteristic of the nozzle. In the view menu, it is possible to arrange images to tiles or cascades. Scaling and erasing of scaling data can be done in the last menu, called others.

On the right panel, there are two edits r0 and r1, where r1 means the maximal radius and r0 means minimal radius. This area is automatically divided to three parts which represent bigger, big and smaller formations of water.

6.3 Opening Image

Pictures of high-pressure nozzles were taken by the Nikon N70 camera and saved in the 16-bit non-compressed Nef format. These pictures were transformed by using the Iris astronomy program, namely its Decode Raw file function, to the FIT or FTS format[1]. Now it is possible to open the file containing the track of the nozzle. We have to check if the file is in correct format by the conditions, so on the 189th position of the file, the number two must be present. Then the 264th and 345th positions contain information about image height and width. So first you have to get information about the size of the picture.

```
While Symbol <> ' ' do
System.Seek(f,269-j);
System.BlockRead(f,Symbol,1);
j:=j+1;
end; //While Symbol <> ' ' do
System.Seek(f,189);
System.BlockRead(f,Symbol,1);
While Symbol <> ' ' do
begin
System.Seek(f,349-i);
System.BlockRead(f,Symbol,1);
i:=i+1;
end; //While Symbol <> ' ' do
for k:=0 to j-2 do
begin
System.Seek(f,269-k);
System.BlockRead(f,Symbols[j-2-k],1);
end;
Parametr:=Symbols[0];
for k:=1 to j-2 do
Parametr:=Parametr + Symbols[k];
OImgWidth:=strtoint(Parametr);
Memo1.Lines[0]:=('Image Widt = ' +Parametr);
for k:=0 to i-2 do
begin
System.Seek(f,349-k);
System.BlockRead(f,Symbols[i-2-k],1);
end; //for k:=0 to i-2 do
Parametr:=Symbols[0];
for k:=1 to i-2 do
Parametr:=Parametr + Symbols[k];
OImgHeight:=strtoint(Parametr);
```

Then, by using function System.BlockRead, we read the data from the file, which starts on position 2880, onto the dynamic variable Img. All operations (Integration and Trasform) use data from this variable. The image in the Img variable is painted on the automatically created image form, the palette is set by the picture gray.bmp.

[1]Which are identical

```
for k:=0 to ImgWidth*ImgHeight - 1 do
begin
x:= k mod ImgWidth;
y:= k div ImgWidth;
Img^[x+y*ImgWidth]:=(Img^[2*x+1 + 2*y*OImgWidth] +
Img^[2*x + 2*(y+1)*OImgWidth]) div 2;
end;
for k:=0 to Imgheight*Imgwidth - 1 do
begin
if Img^[k]>Max then
Max:=Img^[k];
if Img^[k]<Min then
Min:=Img^[k];
end;
Range:=Max-Min;
If Range = 0 then
Range:=1;
GetMem(Line,ImgWidth);
Im.Caption:='Image';
Im.Image1.Width:=ImgWidth;
Im.Image1.Height:=ImgHeight;
Im.Image1.Picture.Bitmap.Width:=ImgWidth;
Im.Image1.Picture.Bitmap.Height:=ImgHeight;
for j:=0 to ImgHeight - 1 do
begin
for i:=0 to ImgWidth - 1
do Line^[i]:=round(255*sqrt((Img^[i+j*ImgWidth]-Min)/Range));
Move(Line^,Im.Image1.Picture.Bitmap.ScanLine[j]^,ImgWidth);
end; // for j:=0 to ImgHeight do
FreeMem(Line);
Im.Image1.Repaint;
```

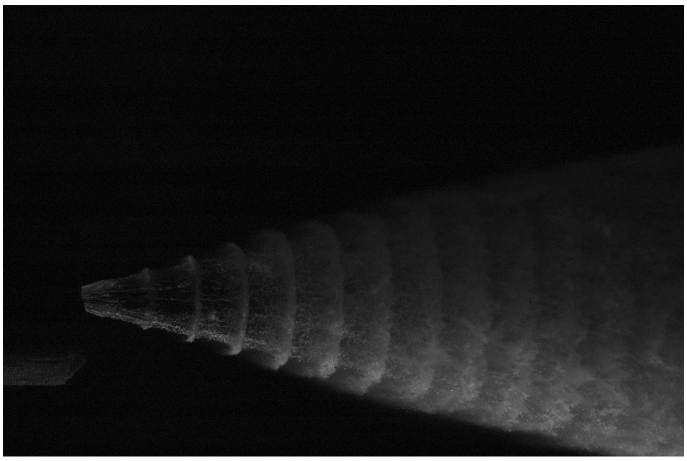

Figure 6.1: Image of the nozzle and its track

6.4 Column Sum

The procedure can be found in the columnsum.pas unit. The Column Sum procedure is used for describing the lengthwise characteristic and for the detection of the nozzle and its water track.

We compute the sum of each column and save them to the Sum variable.

```
for j:=0 to ImgWidth - 1 do
begin
for i:=0 to ImgHeight - 1 do
Sum^[j]:=Sum^[j]+Img^[j+i*ImgWidth];
Sum^[j]:=(Sum^[j]/ImgHeight);
end; //for j:=0 to ImgWidth - 1 do
```

If we want to insert the data into the image, we need to smoothen the data using the Gaussian function.

$$f(x) = \frac{e^{-\frac{x^2}{4\sigma}}}{\sqrt{2\pi\sigma^2}}. \tag{6.1}$$

In this case, $\sigma = 2,5$. From the theory of image analysis we know that, for a 16-bit image, σ is chosen between two and three. This smooth sum is drawn to the image form with a black background. White dots represents the brightness of each column. From this

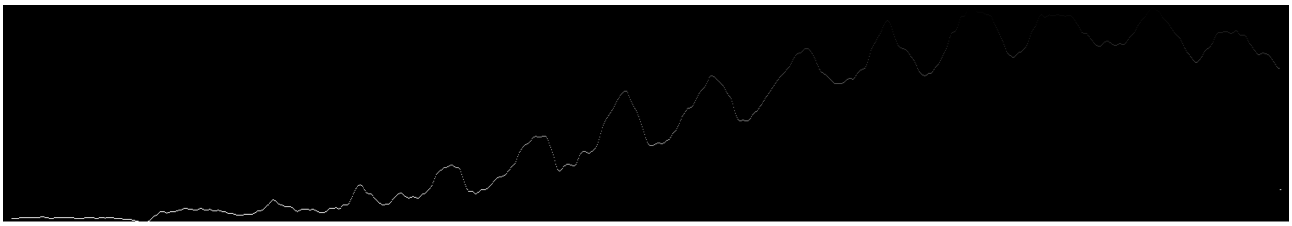

Figure 6.2: Column Sum of the nozzle used in figure 6.1

picture, it is clear that the left almost-constant part is the nozzle and the rest is the water track. Pulses are characterized by the arc. The visibility of the white track displays the fragmentation of water drops. On the right we see the drop more fragmented than at the beginning.

```
for j:=0 to ImgWidth - 1 do
begin
for i:=0 to ImgHeight - 1 do
Sum^[j]:=Sum^[j]+Img^[j+i*ImgWidth];
Sum^[j]:=(Sum^[j]/ImgHeight);
end; //for j:=0 to ImgWidth - 1 do
for i:=10 to ImgWidth - 10 do
begin
NSum^[i]:=0;
for j:=-10 to 10 do
NSum^[i]:=(NSum^[i]+Gauss[j]*Sum^[i+j]);
end; // for i:=10 to ImgWidth - 10 do
```

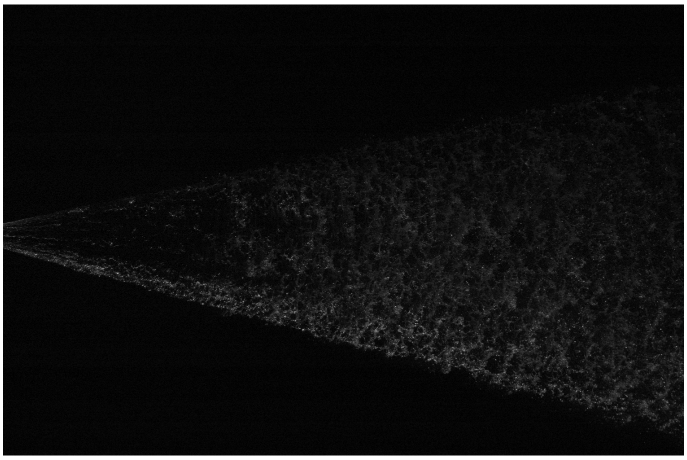

Figure 6.3: Track of the jet without pulsations

On figure 6.3 a nozzle without pulsations is used and on figure 6.4 we see the column sum. It can be seen that the white track is more chaotic than the track on figure 6.2. So it is clear that there are no pulsations.

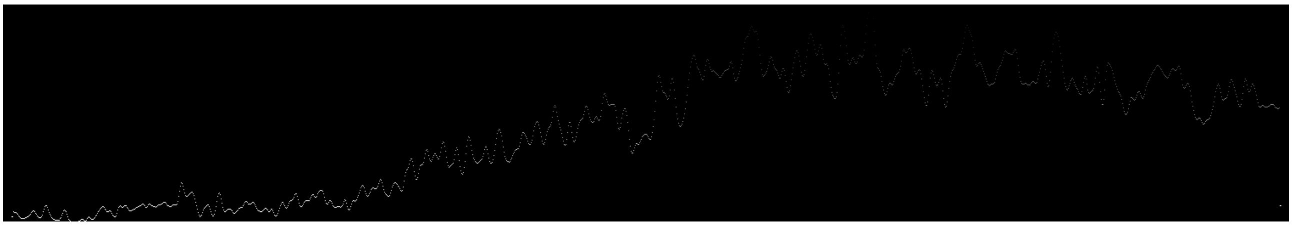

Figure 6.4: Column Sum of the jet used in figure 6.3)

```
Max:=-1000000;
Min:=30000000;
for i:=12 to ImgWidth - 12 do
begin
If Min > NSum^[i] then
Min := (Nsum^[i]);
end; //for i:=10 to ImgWidth - 10 do
for i:=12 to ImgWidth - 12 do
NSum^[i]:=NSum^[i]-Min;
for i:=10 to ImgWidth - 12 do
if Max < NSum^[i] then
Max:=NSum^[i];
Rate:= 255/Max;
for i:=12 to ImgWidth -12 do
NSum^[i]:=(NSum^[i]*Rate);
for i:=0 to ImgWidth*Imgheight -1 do
Picture^[i]:=0;
for i:=0 to ImgWidth - 1 do
begin
```

```
j:=255-round(NSum^[i]);
Picture^[j*ImgWidth+i]:=Round(255 - NSum^[i]);
end;
```

In this procedure, we also see an index between the water track and nozzle. It is calculated as the average of the nozzle part of the column sum. You can see that at the beginning of the column sum, the image is an almost constant track. This track represents the jet in the image. We take the average from this part of the nozzle and try to compute the value of twice the average of the column sum. This value plus five is saved to the index variable.

```
for i:=50 to 129 do
prumer:=prumer+Nsum^[i];
prumer:=prumer/80;
i:=80;
While barva < 2*prumer do
begin
for j:= i to i+9 do
barva:=barva+Nsum^[j];
barva:=barva/10;
Index:=i;
inc(i);
end;
```

The cut image procedure is also declared in this unit. This procedure uses the index value to clip the image.

```
FImgWidth := SImgWidth - Index+5;
FImgHeight := SImgHeight;
for j:=0 to FImgHeight - 1 do
for i:=0 to FImgWidth - 1 do
begin
Image[i,j]:=Img^[i + Index + j * SImgWidth];
end;
```

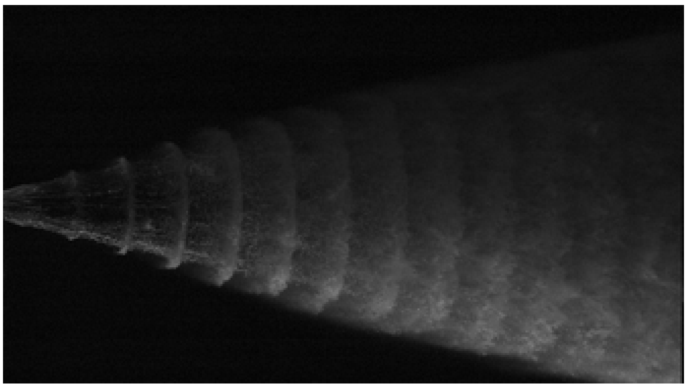

Figure 6.5: Clipped image

6.5 Transform

The Fourier transform has been used in image analysis for many years. There exist many libraries for the Fourier transform. I use the fftw2dll.dll, libfftw.dll and sfftw.dll libraries.

```
function  fftw2d_create_plan( nx, ny, d, f : integer ) : pointer;
          cdecl external 'sfftw.dll';

procedure fftwnd_one( p, i, o : pointer );
          cdecl external 'sfftw.dll';

for  j:=0 to FImgHeight - 1 do
for i:=0 to FImgwidth - 1 do
begin
Spc[i,j].Re:=Image[i,j];
Spc[i,j].Im:=0.0;
end;
pSpc := @Spc;
plan := fftw2d_create_plan( Constant, Constant, FFTW_FORWARD, FFTW_ESTIMATE or FFTW_I
fftwnd_one( plan, pSpc, nil );

b:=Spc;
pSpc := @b;
plan := fftw2d_create_plan( Constant, Constant, FFTW_BACKWARD, FFTW_ESTIMATE or FFTW_
fftwnd_one( plan, pSpc, nil );
for  i:=0 to Constant - 1 do
for j:=0 to Constant - 1 do
ImageM[sn,i,j]:=(b[i,j].Re); //+sqr(b[i,j].Im));
```

The first part of the code is the Fourier transform and the second part is the inverse Fourier transform. This library is defined on a square, so the image is sent to a square matrix with width and height equal to a variable called Constant. This variable is equal to 1536 because this Fourier transform is much faster when the area is divisible by 512.

The amplitude spectrum is computed from the real and imaginary parts of the spectrum.

```
for i:=0 to Constant - 1 do
for j:=0 to Constant - 1 do
begin
AmpSpc[i,j]:=(sqrt(sqr(NSpc[i,j].Re)+sqr(NSpc[i,j].Im)));
end;
```

Logarithmic spectrum is the logarithm of the amplitude spectrum.

```
for i:=0 to Constant - 1 do
for j:=0 to Constant - 1 do
begin
LAmpSpc[i,j]:=(sqrt(sqr(NSpc[i,j].Re)+sqr(NSpc[i,j].Im)));
if  LAmpSpc[i,j]>=0.0000001 then
LAmpSpc[i,j]:=ln(AmpSpc[i,j]);
end;
```

The normal spectrum generated by the Fourier transform has a lower frequency near the point [0,0]. For a better visualization, we translate this point to the centre of the image using the permute function.

```
function Permute(X : integer) : integer;
begin
Permute:=( X - 1 + Constant div 2) mod Constant;
end;

for i:=0 to Constant - 1 do
for j:=0 to Constant - 1 do
begin
ii:=Permute(i);
jj:=Permute(j);
NSpc[i,j].Re:=Spc[ii,jj].Re;
NSpc[i,j].Im:=Spc[ii,jj].Im;
end;
```

Figure 6.6: Image of the Colosseum

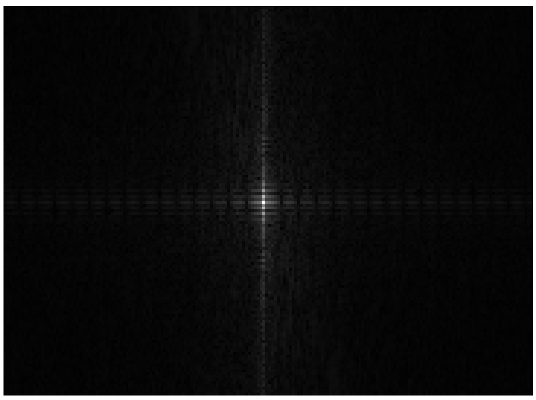

Figure 6.7: Amplitude spectrum of the Colosseum image

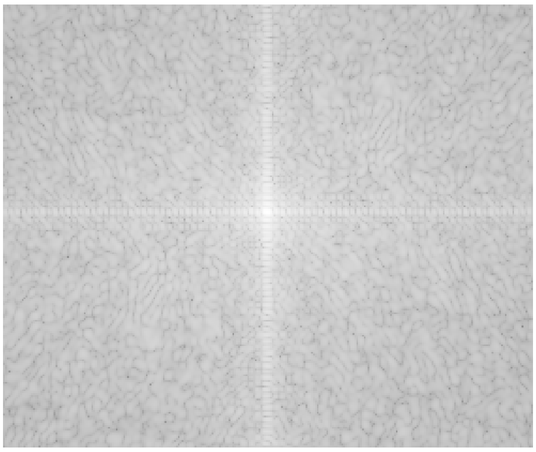

Figure 6.8: Logarithmic spectrum of the Colosseum image

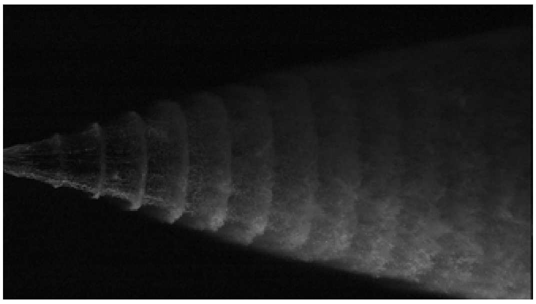

Figure 6.9: Trace of the jet

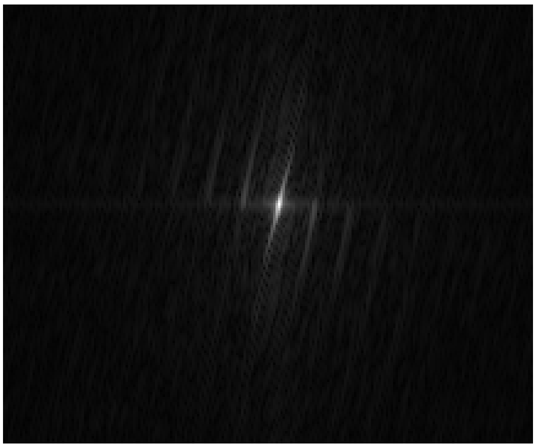

Figure 6.10: Amplitude spectrum of the jet image

The central point in the permuted spectrum (figure 6.10) is not truly important but the arches are important and show that there is something in the direction of the horizontal axis.

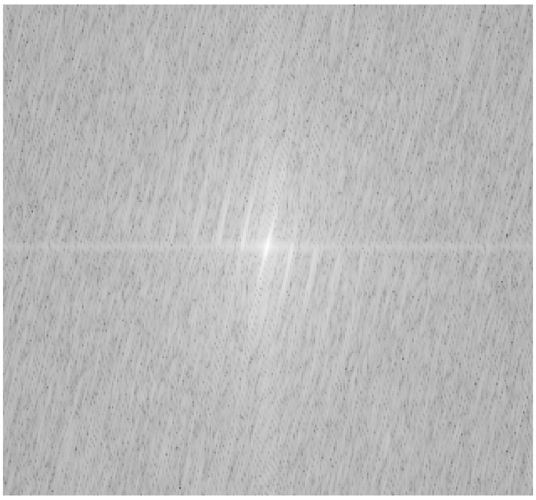

Figure 6.11: Logarithmic spectrum of the jet image

6.6 Integration

This procedure is called integration, since we are computing the integral from the defined annulus by r0 and r1. They represents the minimal (r0) and maximal (r1) radius. The main idea is that we want to exclude certain frequencies. We take this annual area and use the weight function. The range of the annual area is divided into three parts. The distance between the first two parts of the circles[2] is equal to $\frac{1}{6}\,range(r0, r1)$. The second part is $\frac{1}{3}\,range(r0, r1)$ long. The third part is the largest, since the Fourier transform is more sensitive near the central point. We will create a field called matrix, in which a zero is assigned to pixels outside of the annual area. The first and last ten percent of the pixel's values are altered by the sinus function, while the rest of the pixels are multiplied by 1. Now we have to normalize this matrix so that the sum of all values in the matrix is equal to 1. Then we multiply the matrix by the spectrum and compute the Inverse Fourier transform.

```
RozsahMax:=strtofloat(BasicForm.Edit2.text);
RozsahMin:=strtofloat(BasicForm.Edit1.text);
Rozsah:=RozsahMax-RozsahMin;
Deleni:=(Rozsah/6);
PocPol:=trunc(deleni/10);
If PocPol = 0 then
PocPol:= 1;
argsinus:=trunc(2*PocPol/Pi);
if argsinus=0 then
argsinus:=1;
```

[2]We use a non-permuted spectrum

```
GetMem(sinus,PocPol*Sizeof(Real));
for i:=1 to pocpol do
sinus^[i]:=sin(1/argsinus*i);
sn:=0;
for k:=0 to 2 do
begin
FourierTransform(Sender);
Sn:=sn+1;
if k=0 then
begin
r0:=RozsahMin;
r1:=RozsahMin+deleni;
end  //if k=0 then
else
begin
r0:=r1;
r1:=r1+sn*deleni;
end; //else
Integral(Sender);
InverseFourierTransform(Sender);
for j:=0 to Constant - 1 do
for i:=0 to Constant - 1 do
ImageM[sn,i,j]:=InverseImage[i,j];
end; // for k:=0 to 2 do
```

The code for computing the weight function is written in the integration.pas file.

```
for j:=0 to Constant - 1 do
for i:= 0 to Constant - 1 do
begin
r:=sqrt(sqr((i+0.5))+sqr((j+0.5)));
if (r < r1) and (r > r0) then
begin
if r <= (r0+PocPol) then
matrix^[i+j*(Constant)]:=sinus^[round(r-r0) + 1];
if r >= (r1-PocPol) then
matrix^[i+j*(Constant)]:=sinus^[round(r1-r) + 1];
if (r > (r0+PocPol)) and (r < r1-PocPol) then
matrix^[i+j*(Constant)]:=1;
end;
if (r >= r1) or (r <= r0) then
matrix^[i+j*(Constant)]:=0;
end;

Sum:=0;
for i:=0 to (Constant - 1)* (Constant - 1) do
Sum:=Matrix^[i]+Sum;
if sum = 0 then
sum:=1;
```

```
for i:=0 to (Constant - 1)* (Constant - 1) do
Matrix^[i]:=Matrix^[i]/Sum;
for j:=0 to Constant - 1 do
for i:=0 to Constant - 1 do
begin
Spc[i,j].Re := Matrix^[i+j*(Constant)] * SPc[i,j].Re;
Spc[i,j].Im := Matrix^[i+j*(Constant)] * SPc[i,j].Im;
end;
```

Results are shown in different colours. The blue colour is used for larger drops, while the red colour represents medium drops and green represents small drops. Figure 6.15 is composed of a red, green and blue image. These pictures still have a problematic visualisation. The composition of drops is not clear enough to allow use of a better procedure for Image evaluation.

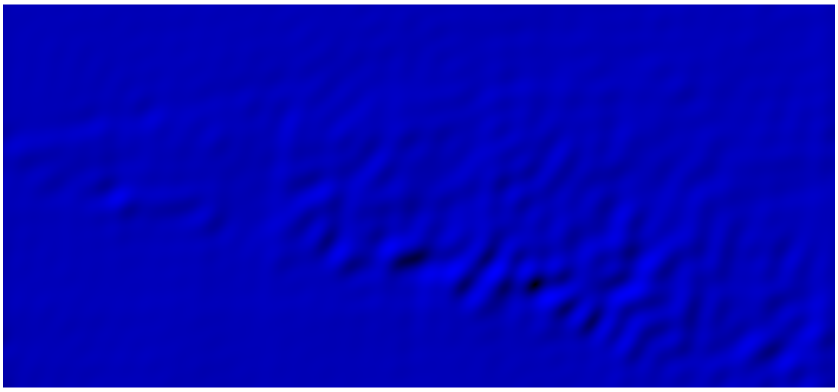

Figure 6.12: Image with bigger drops

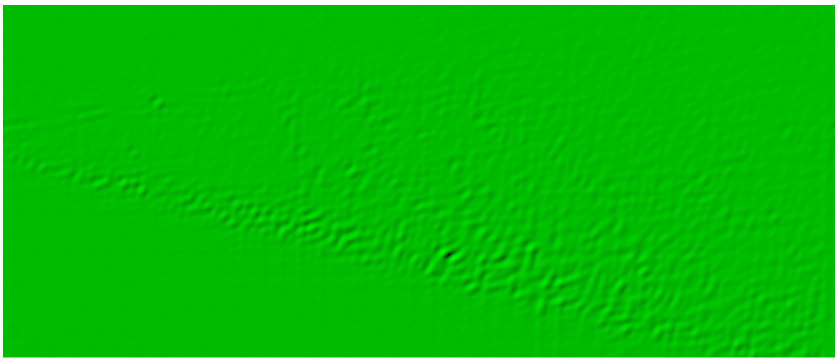

Figure 6.13: Image with medium drops

Figure 6.14: Image with the small drops

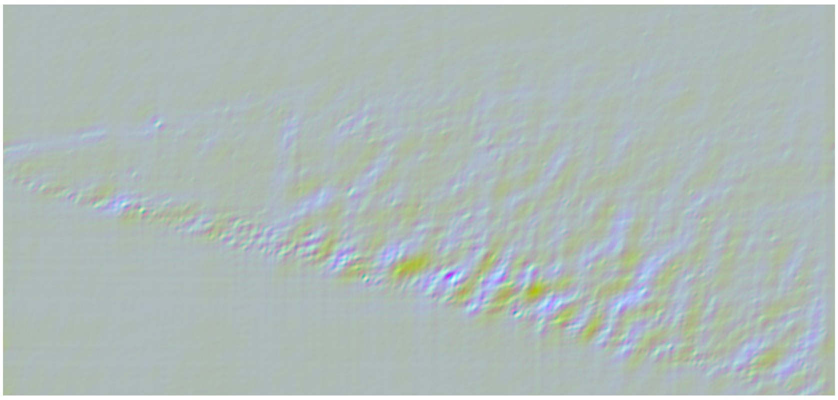

Figure 6.15: Image composed from the red, green and blue images

6.7 Image evaluation

This procedure is similar to integration, however we try to obtain better results. This means that we need to refine the image composed from the red, green and blue image. First we compute the median of each colour in the image. The median is much better than the average because it is not as sensitive to the large differences in the image.

```
for i:=0 to 255 do
for k:=0 to 2 do
histo[k,i]:=0;
medianr:=0;
mediang:=0;
medianb:=0;
for j:=0 to constant - 1 do
for i:=0 to constant - 1 do
for k:=0 to 2 do
begin
inc(Histo[k,round(ImageCol[k+1,i,j])]);
end;   // for k:=0 to 2 do
Max:=0;
for i:=0 to 256 - 1 do
```

```
begin
if max < histo[0,i] then
begin
max:= histo[0,i];
MedianB:=i;
end; // if max < histo[0,i] then
end;  // for i:=0 to 256 - 1 do
Max:=0;
for i:=0 to 256 - 1 do
begin
if max < histo[1,i] then
begin
max:= histo[1,i];
MedianG:=i;
end; // if max < histo[1,i] then
end;
Max:=0;
for i:=0 to 256 - 1 do
begin
if max < histo[2,i] then
begin
max:= histo[2,i];
MedianR:=i;
end;  // if max < histo[2,i] then
end; // for i:=0 to 256 - 1 do
```

Now we will find a constant for each median such that: $medianr = mediang = medinab$. Then we will multiply each colour of the image with the appropriate constant. The colours should be balanced now.

```
konR:=MedianR/MedianB;
konG:=MedianG/MedianB;
for j:=0 to Constant - 1 do
for i:=0 to constant - 1 do
begin
ImageCol[2,i,j]:=ImageCol[2,i,j]*kong;
ImageCol[1,i,j]:=ImageCol[1,i,j]*konr;
end; // for i:=0 to constant - 1 do
```

It is necessary to do a correction of brightness[3]. We use constants for each colour which were generated with respect to the human eye. For the red colour, the constant equals $0,299$. For blue, it is equal to $0,114$ and for green to $wg \;=\; 0,587$. Then we use the following equation

$$jas = (Constant(\,(Red\,wr)\,+\,(Blue\,wb)\,+\,(Green\,wg))) \tag{6.2}$$

to compute the constant, multiply it with each pixel in the image and compute the percentage of overflow. The jas constant is based on this percentage.

[3] In the code it is called jas

```
for j:=0 to constant - 1 do
for i:= 0 to constant - 1 do
begin
kon:=jas/((ImageCol[1,i,j]*wb)+(ImageCol[2,i,j]*wg)+(ImageCol[3,i,j]*wr));
ImageCol[1,i,j]:=kon*ImageCol[1,i,j];
ImageCol[2,i,j]:=kon*ImageCol[2,i,j];
ImageCol[3,i,j]:=kon*ImageCol[3,i,j];
podmin:=true;
for k:= 1 to 3 do
begin
if ImageCol[k,i,j] > 255 then
begin
ImageCol[k,i,j]:=255;
if podmin = true then
begin
prete:=prete+1;
podmin:=false
end;  //if podmin = true then
end; //if ImageCol[k,i,j] > 255 then
end;   // for k:= 1 to 3 do
end;  // for i:= 0 to constant - 1 do
prete:=prete/constant/constant*100;
```

The last step in this procedure is finding the least represented colour in each pixel and the subtraction of sixty percent of this value from all colours in this pixel.

```
for j:=0 to constant - 1 do
for i:=0 to constant - 1 do
begin
medianb:=round(ImageCol[1,i,j]);
mediang:=round(ImageCol[2,i,j]);
medianr:=round(ImageCol[3,i,j]);
if medianb > medianr then
medianb:=medianr;
if medianb > mediang then
medianb:=mediang;
medianb:=round(medianb*0.6);
ImageCol[1,i,j]:=ImageCol[1,i,j]-medianb;
ImageCol[2,i,j]:=ImageCol[2,i,j]-medianb;
ImageCol[3,i,j]:=ImageCol[3,i,j]-medianb;
end;  // for i:=0 to constant - 1 do
```

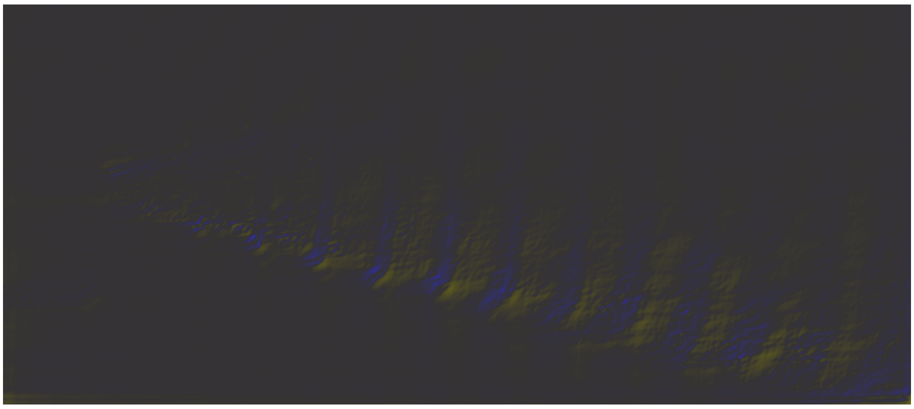

Figure 6.16: Nozzle with pulsations and high speed

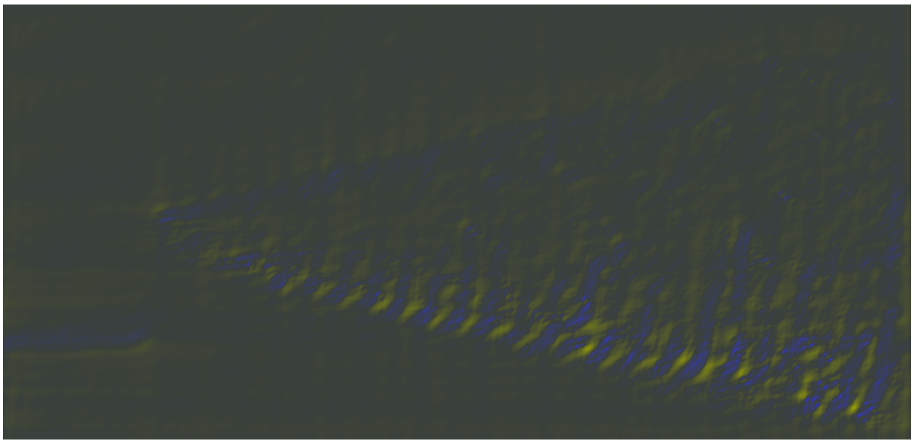

Figure 6.17: Nozzle with the pulsations and medium speed

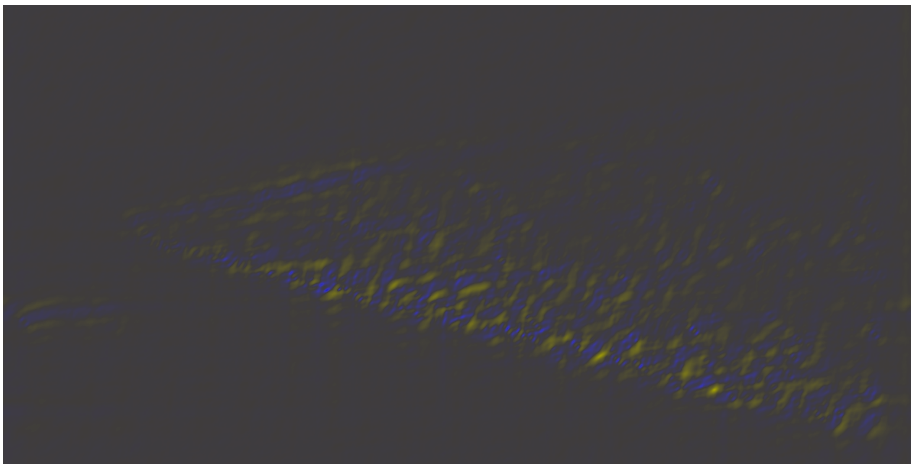

Figure 6.18: Nozzle without pulsations

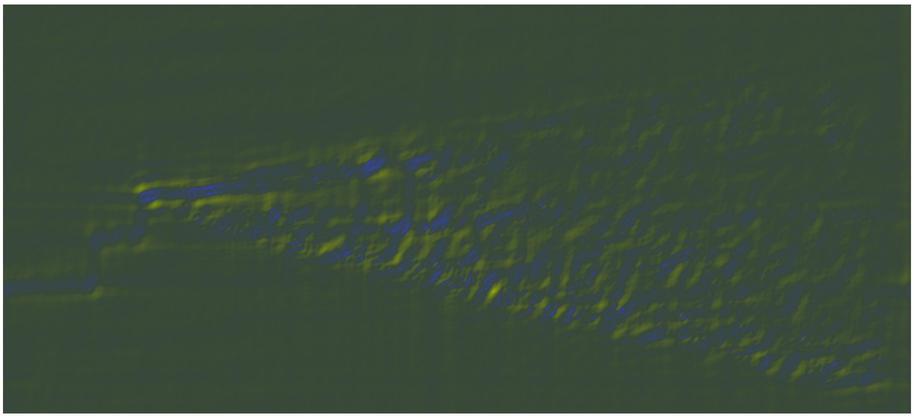

Figure 6.19: Nozzle without pulsations

On these pictures you can see results from the program. The first two pictures (figures 6.16, 6.17) are nozzles with pulsations. You can see that pulses generate bigger formations of the water. At the end of the track, these formations are almost connected. The bigger forms are more chaotic for lower speeds. Non-pulsed nozzles originate from different companies. The jet on the third picture (figure 6.18) produces larger water forms than the fourth one.

6.8 Calibration

It is not possible to distinguish the real size of the object on the photo. Since large models taken from far away could have the same size on the photo as small models taken from close by. For each kind of nozzle we made a special photo of the ruler to allow distance checking on the pictures. All that needs to be done is use of the Scale procedure. Open a file with the ruler and move your mouse to a value on the ruler, then hold the left mouse button and drag the mouse two centimetres before releasing the button. Then click on the Scale procedure in the menu.

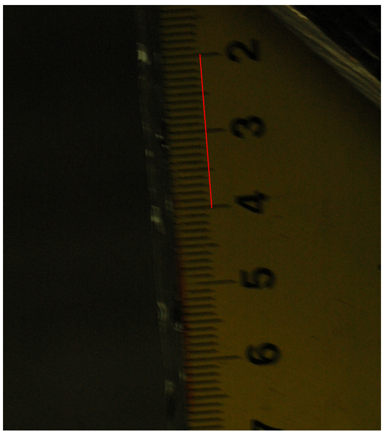

Figure 6.20: Ruler with the red straight line

```
begin
Image3.Invalidate;
If (X>X1p) and (Y>Y1p) then
begin
Image3.Left:=X1p-Lx;
Image3.Top:=Y1p-Ly;
Image3.ClientWidth:=(X-X1p);
Image3.ClientHeight:=(Y-Y1P);
Image3.Canvas.Rectangle(0,0,X-X1P,Y-Y1p);
Image3.canvas.Pen.Color:=clred;
Image3.Canvas.MoveTo(0,0);
Image3.Canvas.LineTo(X-X1P,Y-Y1P);
end;  //If Y>Y1p then
If (X<X1p) and (Y>Y1p) then
begin
Image3.Left:=X-Lx;
Image3.Top:=Y1P-Ly;
Image3.ClientWidth:=(X1P-X);
Image3.ClientHeight:=(Y-Y1P);
Image3.Canvas.Rectangle(0,0,X1P,Y-Y1p);
Image3.canvas.Pen.Color:=clred;
Image3.Canvas.MoveTo(X1P-X,0);
Image3.Canvas.LineTo(0,Y-Y1P);
end; // If Y>Y1p then
If (X>X1p) and (Y<Y1p) then
begin
Image3.Left:=X1p-Lx;
Image3.Top:=Y-Ly;
Image3.ClientWidth:=(X-X1p);
Image3.ClientHeight:=(Y1p-y);
Image3.Canvas.Rectangle(0,0,X-X1P,Y1p-y);
Image3.canvas.Pen.Color:=clred;
Image3.Canvas.MoveTo(0,Y1p-y);
Image3.Canvas.LineTo(X-X1p,0);
end; //If Y<Y1p then
If (X<X1p) and (Y<Y1p) then
begin
Image3.Left:=X-Lx;
Image3.Top:=Y-Ly;
Image3.ClientWidth:=(X1P-X);
Image3.ClientHeight:=(Y1p-Y);
Image3.Canvas.Rectangle(0,0,X1P,Y1p-y);
Image3.canvas.Pen.Color:=clred;
Image3.Canvas.MoveTo(X1P-X,Y1p-Y);
Image3.Canvas.LineTo(0,0);
```

X and Y is the current position in the image, X1P, Y1P is the starting position of movement and X2P, Y2P ihes t final position. Lx and Ly represent the position of the scrollbar.

The scale procedure will compute a scale variable called meritko. If you wish to erase this variable, just click on the erase data procedure;

```
vzdalenost:=sqrt(sqr(x2P-x1P)+sqr(Y2P-Y1P));
If vzdalenost<>0 then
meritko:=20/vzdalenost;
```

Chapter 7

Conclusion

The first problem was to transfer images to an easily readable format. At first we tried using the tiff format, but during processing it was discovered that this format was not suitable because different programs made different pictures. So it was necessary to find and use something else. The best results were obtained using the Iris astronomy program for transfering pictures to the fit or fts format. It was easy to load pictures from this format, simply because the format is not too complicated. Once the image is loaded, it is possible to use the Fast Fourier transform. This is the most important part of the work. The image is transformed to its spectrum and by using certain weighted matrices it is possible to characterize jets, which are used for cooling materials and removal scales. The theory states that a nozzle with the coarser structure is better for removal scales. We used a special delphi library for the direct and inverse transforms. It wouldn't make much sense to try to program this transformation on our own, because the program is quite complicated and it would not be as fast. Now it was necessary to cut certain frequencies. We obtained an image only containing the data we were interested in from the filtered spectrum. It is not simple to properly set the parameters for an optimal result. A smart choice of radii is between five and two hundred and fifty, depending on the image. Usually, frequencies larger then five hundred can be safely considered noise. The fact that the track of the jet is on the lower side means that having more light is not important, since it will affect only very low frequencies. Three colours are used for visual representation. The blue colour symbolizes bigger drops, green represents medium drops and red is for small drops. Of course this depends on the choice of radii. For better visual representations, an image evaluation method is used. This procedure produces a coloured image where it is possible to see differences between drops. The required brightness needs to also be set. When the overflow is too large, it is necessary to decrease the brightness constant. This allows comparison of nozzles from different companies.

Bibliography

[1] *Druckmüller, M., Hariban, P.:* **Digital Image Processing System for Windows version 5.0.**,
Sofo, Brno 2004

[2] *Bezvada, V.,Ježek, J.,Saic, S.,Segeth, K.:* **Dvojrozměrná diskrétní Fourierova transformace a její použití**,
I.Státní pedagogické nakladatelství, Praha 1988

[3] *Štarha, P.,Druckmüller, M.:* **Adaptive contrast control, Image Structure and Object Analyser manual**

[4] *Fiřt, J.,Holota, R.* **Digitalizace a zpracování obrazu**,
`http://home.zcu.cz/ holota5/publ/DigZpr0.pdf`

[5] `http://www.wikipedia.org`

[6] `http://sumitbirla.com/software/imagepro2/index.php`